REMARKS

Claims 37-44, 46-50, 52-58, and 60 were pending and rejected. In response, Applicant have amended claims 37, 38, 43, 44, 46-50, 52-55, 57-58 and 60 as set forth above. Claims 1-36, 45, 51 and 59 were previously cancelled. No new matter has been introduced. Accordingly, claims 37-44, 46-50, 52-58, and 60 are now pending.

Claims Objections

Claims 37, 38, 47, 53-55, 57 and 60 were objected to for various informalities. The objections have been addressed by the above amendments. Thus, withdrawal of the objections is respectfully requested.

35 USC §101 Rejections

In the subject Office Action, claims 47-50 and 52-55 were rejected under 35 USC §101. In response, Applicant has amended 47-50 and 52-55, overcoming the Examiner's rejection. Withdrawal of the rejection is respectfully requested.

35 USC §103 Rejections

In the subject Office Action, claims 37-39, 43-44, 46-48, 50, 52-58, and 60 were rejected under 35 USC §103(a) as being unpatentable over Requa ("The Piecewise Data Flow Architecture: Architectural Concepts,") in view of Pattern et al. ("Computer Architecture, A quantitative Approach.). In response, Applicant has amended the claims as set forth above, without prejudice. Among other things, amended claim 1 now recites in pertinent part:

"assigning a group of instructions selected from a plurality of groups of instructions partitioned from a program, to a subset of interconnected computation nodes preselected from a plurality of interconnected computation nodes, the instructions having respective associated operands, and each computation node includes a store and an execution unit having one or more arithmetic logic units, floating point units, memory address units, or branch units;

loading a subset of instructions of the assigned group of instructions into a frame of buffers comprising the stores disposed on the preselected subset of interconnected computation

nodes having been assigned the group of instructions, wherein the loading is performed prior to the associated operands of the subset of instructions are available ..."

In the subject Office Action, when rejecting claim 37, the Examiner cited Requa as teaching or suggesting the "assigning ..." recitation, and Requa in conjunction with Patterson as teaching or suggesting the "loading ..." recitation. In response, notwithstanding Applicants respectful disagreement with the Examiner's reading of the claim and the references, Applicants have amended Claim 37 to recite "each computation node includes a store and an execution unit having one or more arithmetic logic units, floating point units, memory address units, or branch units." Requa disclosed a data flow architecture that includes a Block Processing section, an Instruction Issue section and an Instruction Processing section having a number of scalar preprocessors, a memory processor and a SIMD processor. As described in the section cited by the Examiner, under Requa "Instructions are grouped (by the Block Processing section) into relatively small blocks" and dispatches on a block by block basis to the Instruction Issue section, which in turn issues the instruction to one or more of the scalar processors, the memory processor or the SIMD processor to execute, as the operands of the instructions become available. Under Requa, there is no concept at dispatch time that the block of instructions is "assigned" to be executed by a subset of the scalar, memory and SIMD processors. Which processor is ultimately employed to execute each instruction, and thus the combination of processors employed to execute a block of instructions, is decided by the Instruction Issue section (based on the instruction type). Accordingly, even if we ignore the fact that Requa's scalar, memory and SIMD processors are not "computation nodes" as each of the scalar, memory and SIMD processors do not includes "... a store and an execution unit having one or more arithmetic logic units, floating point units, memory address units, or branch units," Applicants submit Requa nonetheless fail to teach or suggest the "assigning a group of instructions selected from a plurality of groups of instructions partitioned from a program, to a subset of interconnected computation nodes preselected from a plurality of interconnected computation nodes, ..." of claim 37.

In the subject Office Action, when rejecting claim 37, the Examiner also reasoned that Requa in combination with Patterson teaches or suggest the "loading ..." recitation, as Requa teaches "loading the instructions ...," except for the fact that Requa's instruction buffers to store the block of dispatched instructions reside in the Instruction Issue section , as opposed to the

functional units, but that is allegedly remedied by the reservation stations teaching of Patterson. First of the all the "loading" clause of claim 37 recites "loading a subset of instructions of the assigned group of instructions ..." Given that Requa does not teach or suggest the "assigning ..." recitation as discussed earlier, it follows then Requa does not teach of suggest "loading a subset of instructions of the assigned group of instructions ..." regardless of Requa's buffer structure, and whether Pattern adequately remedies the admitted deficiencies of Requa. Further, even if we are to ignore the fact that there is no notion or suggestion in Requa and/or Patterson about "loading a subset of instructions of the assigned group of instructions ...," the reservation stations of Patterson are disposed within the functional units, the ALU, the multiplier and so forth. So in combination, Requa and Patterson, at most suggest the loading of a block of instructions into a frame of buffers disposed across a number of function units, ALU, multiplier and so forth, but the combination does not suggest the recitation of "loading a subset of instructions of the assigned group of instructions into a frame of buffers comprising the stores disposed on the preselected subset of interconnected computation nodes having been assigned the group of instructions ..." where "each computation node includes a store and an execution unit having one or more arithmetic logic units, floating point units, memory address units, or branch units."

Accordingly, for at least the foregoing reasons, Applicants submit amended claim 37 is patentable over Requa and Patterson, individually or in combination, 35 USC §103(a).

Claims 47, 57 and 60 have been similarly amended as claim 37. Thus, for at least similar reasons, claims 47, 57 and 60 are patentable over the cited references.

Claims 38-44, 46, 48-50, 52-56, and 58 depend from either claim 37, 47 and 57, incorporating their recitations. Therefore, for at least similar reasons, claims 38-44, 46, 48-50, 52-56, and 58 are patentable over the cited references. Claims 38-44, 46, 48-50, 52-56, and 58 are further patentable over the cited references by virtue of their additional recitations.

Conclusion

In view of the foregoing, Applicant submits that, all remaining pending claims are now in condition of allowance. Therefore, early issuance of a Notice of Allowance is respectfully requested. Should there be any question with Applicants' response, Applicants invite the Examiner to contact the undersigned at 206-381-8819 (Direct).

Lastly, the Commissioner is hereby authorized to charge shortages or credit overpayments to Deposit Account No. 500393.

Respectfully submitted,
Schwabe, Williamson & Wyatt, P.C.

Dated: June 3, 2010

/Al AuYeung/
Al AuYeung
Reg. No. 35,432

Schwabe, Williamson & Wyatt P.C.
1420 5th Avenue, Suite 3400
Seattle, Washington 98101
Telephone: (206) 622-1711
Facsimile: (206) 292-0460